

APPLICATION FOR
UNITED STATES LETTERS PATENT
SPECIFICATION

Inventor(s): Makoto SUNADA

Title of the Invention: APPARATUS FOR SAVING FILES INTO
A PLURALITY OF HOLDERS

09750891-01401
TQ2TQ T8809260

APPARATUS FOR SAVING FILES INTO A PLURALITY OF HOLDERS**Background of the Invention****Field of the Invention**

5 The present invention relates to an apparatus for saving files into a plurality of saving destinations.

Description of the Related Art

10 Owing to the recent technical development of information apparatuses and network information communication fostered by the Internet which has been developed in association therewith, many users have come to open their homepages. In the preparation of the homepage, after the homepage has been hierarchically
15 configured to prepare each page for each hierarchy, the homepage is saved into a specific saving destination and then a link from the top page is prepared. As described above, there has been adopted a conventional system in which when a file was to be saved, the file
20 was always surely saved into one saving destination under one unique name, so that it was necessary for an implementor of a homepage to save the file for each page into an appropriate saving destination after the file of each page had been prepared. However, in a case
25 that the homepage is constituted by many low-layered

09760831-01101

pages, after each page has been prepared, files for these many pages should be appropriately copied in sequence and the adjacent pages should be linked together.

That is, in the prior art, the file is saved at one location and the user copies and moves the file as required. For example, in case of saving of public data such as HTML files or the like for the homepage, a file which is under preparation is not made public, but the file is made public in a state that it is finished as its final form. In such a case as mentioned above, conventionally, the file was updated in a working directory to be finished as its final form and thereafter the file was copied (moved) into a public directory.

Summary of the Invention

An object of the present invention is to provide a file saving apparatus which is simple in structure and is capable of saving files into a plurality of saving destinations.

A file saving apparatus according to the present invention comprises plural saving designating means for designating file saving destination at a plurality of saving destinations and file saving means for saving the files at the plurality of saving destinations thus designated.

According to the present invention, since the files can be stored at the plurality of saving destinations at one time, failure to copy and move the file or the like can be prevented and hence the working efficiency can be improved in the preparation of the homepage or the like.

Brief Description of the Drawings

Next, the present invention will be described in more detail with reference to the accompanying drawings. In the drawings,

FIG. 1 is a diagram showing the structure of a system according to an embodiment of the present invention;

FIG. 2 is a flowchart illustrating a flow of a process relating to the embodiment of the present invention in the system shown in FIG. 1;

FIG. 3 is a flowchart illustrating a flow of a saving condition judging Process;

FIG. 4A and FIG. 4B are diagrams showing examples of file saving information and file saving condition tables;

FIG. 5A and FIG. 5B are diagrams showing other examples of file saving information and file saving condition tables;

FIG. 6A and FIG. 6B are diagrams showing further

5

10

15

20

25

FIG. 14 is a diagram showing a concrete example of a condition table within the saving condition storage area shown in FIG. 13B;

FIG. 15 is a diagram showing a concrete example of inter-link information of related files; and

FIG. 16 is a diagram illustrating a hardware environment for the information apparatus required when

the embodiment of the present invention is implemented as a program.

Description of the Preferred Embodiment

5 In the embodiment of the present invention, a plurality of saving destinations, including a public directory and a plurality of saving conditions, can be designated and upon file saving, files can be saved at a plurality of saving destinations at one time, so that
10 the necessity to copy (move) the file can be eliminated.

 In addition, operations conducted after the file has been saved are automated, so that the failure to save the file can be prevented and hence the working efficiency can be improved.

15 Fig. 1 is a diagram showing the structure of a system according to an embodiment of the present invention.

 The system according to this embodiment is constituted by an information apparatus such as a computer 10 or the like. The computer 10 comprises a file
20 system 11 and an executing device 17. In general, an application 14 is stored in the file system 11 before it is started and is loaded to the executing device 17 when it is to be started. The file system 11 is constituted by a hard disk, CD-ROM, DVD, a floppy disk or the like.
25 And, the executing device 17 is constituted by a ROM,

a RAM, a CPU or the like.

In addition to the application 14, the executing device 17 includes a plural saving information memory device 16 for storing information at a location the file is to be saved. The plural saving information memory device 16 may be implemented either by providing a dedicated memory in the executing device 17 to store plural file preserving information therein or by storing the plural file saving information at the file system 11 so as to load the information from the file system 11 to the RAM or the like of the executing device, as required.

The application 14 acquires the plural file saving information from the plural saving information memory device 16 and the file prepared by the application 14 is stored into a plurality of storage locations in the file system 11 as a saved file 13. In this case, the plural file saving information is stored in the plural preserving information memory device 16, which is constituted by the dedicated memory provided in the executing device 17, or is stored in a specific area 16 within the file system 11.

Fig. 2 is a flowchart illustrating a flow of a process relating to the embodiment of the present invention executed in the system shown in Fig. 1.

First, it is supposed that there exists a file to be stored into a plurality of saving destinations in a state that the application is started.

In this case, at step S10, a plural file preserving mode is selected. Then, at step S11, saving destination information is read in. At step S12, dialogue for designating a file saving destination is displayed and, then, at step S13, to which saving destination the file is to be saved is designated by a user.

Then, at step S14, the file is saved in one of the saving destinations designated by the user as the first file based on the information designated by the user at the step S13. At that time, link information to the plural saving information memory device is also saved as file information. Next, a plural saving process is started from step S15.

First, at step S16, a preserving condition (which will be described later) is judged and, at step S17, the first file which has been saved first is copied to another saving destination which is designated by the user. After the first file has been copied in the above mentioned manner to all of the saving destinations designated by the user, the plural saving process is finished at step S18. Then, at step S19, the saving destination information is saved and, at a step S20, completion of

the plural saving process is indicated to the user on a pop-up screen to finish the process.

Fig. 3 is a flowchart illustrating a flow of a saving condition judging process.

5 First, at step S25, saving date, file size and implementor information are acquired from the first file which has been saved first. Then, at step S26, in a case that there exists a saving destination file (that is, in the case of addressing to an existing file), the saving date, the file size and the implmentor information are
10 acquired from this saving destination file.

Then, at step S27, based on the file saving conditions, the saving file information and saving destination file information thus acquired are compared with each other
15 to judge whether the file saving process is executed. As a result, when it is judged that the file saving process is executed, the flow goes to step S17 in Fig. 2. On the other hand, when it is judged that the file saving process is not executed, the process is finished.

20 Figs. 4A and 4B show examples of file saving information and file saving condition tables.

As shown in Fig. 4A, the file name, the preserving condition and the file date are registered as the file saving information. In the case of the example shown
25 in Fig. 4A, an entry (0) of the file saving

information indicates the information on the original saving file (the first file which has been described above) and it can be seen that the file date is 2000/05/04. The entries (1) to (3) of this file saving information indicate information on files to be stored at a plurality of saving destinations, and respective file names and file dates are registered and the presence or absence of a designation of a saving condition is registered. In the case of the example shown in Fig. 4A, the saving conditions are designated for all of the plurality of saving files.

Fig. 4B shows the examples of the file saving condition tables and each saving condition table is linked with each entry of the file saving information. For example, the date information is designated in the saving condition table corresponding to the entry (1) of the saving information shown in Fig. 4A as a saving condition item, that is, such a condition is designated that a file dated earlier than 2000/05/01 should be stored. Likewise, the date information is designated in the saving condition table corresponding to the entry (2) of the saving information as an item, that is, such a condition is designated that a file dated earlier than 2000/05/02 should be saved. Then, the date information is designated in the saving condition table

corresponding to the entry (3) of the saving information, that is, such a condition is designated that a file dated after 2000/05/02 should be saved.

Accordingly, all of the plurality of saving files
5 of the entries (1) to (3) registered as shown in Fig. 4A conform to the conditions of the saving condition tables in Fig. 4B and hence all of the plurality of saving files are saved.

As has been described above, in a case that a file
10 (such as HTML file of a homepage, a source code file of a program or the like) -presupposing that the same file is always updated- is to be saved, a condition that a file of a certain period (date) is saved is designated.

In this case, it becomes possible to sequentially
15 save respective files which are being worked into a separate directory with no need to use a specific history management tool by designating time, thereby the file can be readily brought back to its original state and problems can be solved at an early stage.

Accordingly, in the case of the example shown in
20 Fig. 4B, files with dates earlier than 2000/05/01, earlier than 2000/05/02 and after 2000/05/03 are saved as the entries (1) d:\rikreki\src.java, (2) d:\rikreki\2src.java, and (3) d:\rikreki\3src.java,
25 respectively, depending on the filing dates under which

the files concerned are stored.

Figs. 5A and 5B are diagrams showing other examples of file saving information and a file saving condition table.

5 In Fig. 5A, as contents to be registered, the file name, the saving condition and the file size are designated and the original saving file of the file size of 1.5MB and a plurality of saving files of the file size of 0.9MB and for which the saving conditions are
10 designated are registered.

 In the saving condition table corresponding to the entry (1) of the saving information in Fig. 5A, the file size is set as an item as shown in Fig. 5B. In this case, such a condition is designated that the file of the file
15 size of 0.5MB or less should be preserved, which means that the file of the entry (1) in Fig. 5A has a file size of 0.9MB and hence is not saved.

 In the case of preparation of a homepage, it may sometimes occur that the quantity is fixedly limited
20 by a provider with which the user made a contract or that a large-size file should be divided into sections by taking the download time into consideration. Because of a limitation in file size, when an error occurs in the condition, the file concerned
25 is not saved in the directories of the plurality of saving

destinations, but error information is messaged. As a result, it becomes possible for the user to recognize the necessity to change the file size when the user intends to preserve the file and hence the possibility of avoiding problems concerning the file size can be increased.

This example shows that, upon plural file storage, the file of (1) d:¥sizecheck¥mydata.html cannot satisfy the saving condition in term of the file size and a message that the file could not be saved is issued at the end of the holding process.

Figs. 6A and 6B are diagrams showing further examples of the file saving information and file saving condition table.

In the saving information shown in Fig. 6A, as the contents to be registered, the file name, the saving condition and the implementor are designated, and an original saving file prepared by a person named SUZUKI is registered in the entry (0) and a plurality of saving files also prepared by SUZUKI are registered in the entry (1). Saving conditions are designated for the plurality of saving files.

Fig. 6B shows the example of the file saving condition table, the implementor information is registered as an item and "SUZUKI", which is the name of the implementor,

is registered as a condition. Accordingly, in a case that saving corresponding to the entry (1) of the saving information shown in Fig. 6A is to be conducted, the saving condition table in Fig. 6B is referenced. In this case, the name of the implementor is SUZUKI and the saving information coincides with the condition in the saving condition table, so that the file of the entry (1) of the saving information is saved.

In a case that a plurality of persons intend to change one common file on a server in the course of group or joint development or the like, file storing destinations are determined in advance for individual implementors in the saving directory, by which file replacement by illegal access can be checked and secure file saving by the implementor can be realized.

In the case of the examples shown in Figs. 6A and 6B, upon plural file storage, the file of (1) d:¥suzuki¥mydata.html satisfies the condition and hence it is to be saved.

As has been described above, an erroneous operation by the user in the course of routine saving work and a failure to save the file can be prevented by setting the saving condition, by which such advantages can be attained that there is no need for the user to memorize the saving location and the execution time can be

shortened.

Fig. 7 is a flowchart illustrating a flow of a process of acquiring a plurality of saving destinations.

First, at step S30, link information linking to the plural saving information memory device is acquired from a file or the plural saving information is retrieved by using the file name as a key to acquire the link information. Then, at step S31, it is checked to determine whether the link information is present. When the link information is absent, the process is finished. On the other hand, when the link information is present, the next piece of link information is acquired from the plural saving information at step S32 and it is utilized in the application. Then, at the completion of utilization of the next piece of link information, the flow goes back to the step S31 and the process is repeated as long as the link information is present.

Next, an example of a method of designating plural file saving in which the embodiment of the present invention is used will be described.

Fig. 8 is a diagram exemplifying a screen displayed when a plural saving mode is selected.

The plural saving mode is configured such that the user can select the mode, for example, from a pull-down menu in a window. In the case of the example shown in

Fig. 8, "PLURAL SAVING" is selected from "FILE" in the pull-down menu. Then, a plural saving designation screen is displayed and a holder file name, an attribute of the file to be saved and a saving condition are respectively designated for the plural saving. Then, the file is saved and a result thereof is issued.

Fig. 9 is a diagram exemplifying a screen for plural file saving designation. On the screen, there exist areas for designating a file saving destination, attributes of the file and saving conditions.

The user designates the saving destination of the original file and the saving name by designating the saving destination and the file name in an upper column on the plural saving designation screen. The dialogue in a lower column on the plural saving designation screen is an area for designation when plural file saving is executed.

In the example shown in Fig. 9, the information on the original file is displayed on this dialogue in the beginning. Next, the saving file name of the file for the plural preservation is designated. This saving file name is described starting with the drive name and together with the saving location. The user enters the dialogue by selecting a button for setting ADD, CANCEL or SAVING CONDITION displayed on the right-hand side

of the dialogue to conduct setting operations required for the plural saving. ATTRIBUTE in the dialogue concerned is the one for which a read-only type one, an archival type one or the like which is generally known as the attribute of the file is set and the attribute to be set is displayed when the file is saved. CONDITION means the above mentioned saving condition for judging whether the file is saved by referrencing the saving condition table.

10 Fig. 10 is a diagram exemplifying a screen for a notice of completion of file storage.

As shown in Fig. 10, the saving file name, marks indicative of whether the file corresponding to this name has been successfully saved and reasons for failure when the file has not been successfully saved are displayed on the screen. The user may go back again to the plural saving designation screen with respect to the file which has not been successfully saved to again make the setting such that the file can be successfully saved or to end the plural file saving process judging that unsuccessful file saving is valid.

Fig. 11 is a diagram showing concrete examples of file definition information used on the screen shown in Fig. 9.

25 The file definition information is that defined for

00760884 04794

each file and the file name, the updating date, the size and the implementor of the file to be pluraly saved are stored therein. Upon the plural saving, the saving condition is examined in accordance with this definition so as to save only a file corresponding to the saving condition. In the example shown in Fig. 10, the file of the entry (1) in Fig. 11 was not saved. The reason lies in the fact that the size of the file is larger than the value set in the saving condition. For example, in the saving condition table, the file size is set to be 1MB or less as the saving condition. In a case that the saving condition as mentioned above is set, the file of the entry (1) in Fig. 11 is 1.2MB in size and hence is not saved.

Fig. 12 is a diagram exemplifying a configuration of a plural saving information recording device.

There exists a plural file saving information memory device in the filing system, as shown in Fig. 12, for the plural file saving designation screen shown in Fig. 9. In this connection, it is to be noted that although the word "device" is used, this may simply be a memory area for storing information. The plural file saving information memory device is provided with a sufficient number of areas for storing all of file saving destinations, file attributes and saving

conditions which would constitute plural file saving information. In this example, the file saving destinations, the file attributes, the saving conditions and inter-link information are stored as the plural file saving information. In the plural file saving information memory device, a file attribute is assigned to the saving destination of the file corresponding to the information thus stored, and when the file conforms to the saving condition, the file is saved, and then the file thus saved is copied tracing the next piece of file saving information based on the link information and the file thus copied is saved as the same file as the above.

Figs. 13A and 13B are diagrams exemplifying configurations of file attribute information storage areas and a saving condition storage region.

Fig. 13A exemplifies the file attribute storage area. In this example, the area is obtained such that READ-ONLY, COVERT FILE, ARCHIVE and SYSTEM can be set as the file attributes. Fig. 13B exemplifies the saving condition storage area. In this example, the area is constructed such that the holding conditions in which DATE INFORMATION, FILE SIZE and IMPLEMENTOR INFORMATION are used as keys can be set.

Fig. 14 is a diagram showing a concrete example of

a condition table within the saving condition storage area shown in Fig. 13B.

The concrete example shown in Fig. 14 is configured such that DATE INFORMATION, FILE SIZE and IMPLEMENTOR INFORMATION can be registered as items. In the case of DATE INFORMATION, such a concrete condition is set that a file of a date after 2000/05/01, 10:00:00 and earlier than 2000/05/10, 0:00:00 should be saved. In addition, such conditions are also set such that when the file size is 1 MB or less and the implementor is "SUZUKI" or, in Chinese characters, "鈴木", the file should be saved.

Fig. 15 is a diagram showing a concrete example of inter-link information of related files.

The inter-link information is the one for managing link information linking together a plurality of files designated to be saved when plural saving is designated and is set when the files are to be saved.

Fig. 15 shows the example of storage information when the plural saving information is treated as one file. In order to acquire the plural file saving information, the inter-link information indicative of the plural saving information is stored. That is, in the case of the example shown in Fig. 15, when a file such as "e:¥linkdata¥linkdata1.dat" is pluraly

saved, "e:\linkdata\linkdata2.dat" is related thereto as a file related to the above such that the related files which have been plurally saved can be readily acquired. Likewise, as files related to the
5 file "e:\linkdata\linkdata2.dat", "e:\linkdata\linkdata1.dat" and "e:\linkdata\linkdata3.dat" can be readily acquired. Accordingly, when a homepage or the like is prepared in the application, the related files can be readily found and, in addition, failure to update
10 a file which should be updated can be prevented.

Fig. 16 is a diagram illustrating a hardware environment of an information apparatus needed when the embodiment of the present invention is implemented as a program.

15 The information apparatus 31 comprises a CPU 21, a ROM 22, a RAM 23, a communication interface 24, a recording device 27, a recording medium reading device 28, and an input/output device 30 which are inter-connected via a bus 20. The program for
20 implementing the embodiment of the present invention can be stored in the ROM 22. Although, the ROM 22 is typically adapted to store a basic program for realizing data exchange between the input/output device 30, consisting of a display, a mouse, a keyboard and the like
25 such as BIOS or the like, and the CPU 21, the program

for implementing the embodiment of the present invention may be stored in the CPU 21 so as to use the information apparatus 31 as a special purpose apparatus. In general, the program concerned in the embodiment of the present invention is recorded in the recording device 27, such as a hard disk or the like, and is modified and stored by the CPU 21 in the RAM 23 so as to be brought into an executable state.

As an alternative, the program concerned may be stored in a portable recording medium, such as the CD-ROM, the DVD, the floppy disk or the like, and then is read by a recording medium reading device 28 so as to be stored by the CPU in the RAM 23 such that it is brought into the executable state. Typically, the program concerned, which is read out from the portable recording medium 29, is installed in the recording device 27 to be executed when needed.

As another alternative, the program concerned may be executed by using the communication interface 24 of the information apparatus 31 to connect the information apparatus to an information provider 26 via a network 25 so as to download the program concerned from the information provider 26. Instead, it is also possible to execute the program concerned under the network environment in a case that the network 25 is

5

10

15